Smart Sorting - Transfer Learning for Identifying Rotten Fruits and Vegetables

1. INTRODUCTION

1.1 Project Overview

Smart Sorting is an innovative project focused on enhancing the precision and efficiency of detecting rotten fruits and vegetables using cutting-edge transfer learning techniques. By leveraging pre-trained deep learning models and adapting them to specific datasets, this system provides an automated solution for sorting produce, reducing manual errors, and optimizing quality control in agricultural and food sectors.

1.2 Purpose

- Develop a high-accuracy deep learning model for classifying fresh vs rotten produce.
- Implement a web-based application for real-time classification.
- Support industries with a reliable and efficient tool for food quality management. [Team Member] Team Member: Anuhya.TLV

2. IDEATION PHASE

2.1 Problem Statement

Manual sorting of fruits and vegetables is inefficient, error-prone, and not scalable. Automating the freshness detection process using computer vision can save time, reduce waste, and improve overall product quality.

2.2 Empathy Map Canvas

Think & Feel: Wants efficiency and precision

Hear: Complaints about waste or spoilage

See: Manual labor and sorting lines

Say & Do: Inspects manually, rejects visibly spoiled items

Pain: Time-consuming, error-prone, fatigue

Gain: Faster, automated, reliable solution

2.3 Brainstorming

Initial ideas included:

- Conveyor belt integration with real-time scanning
- Smart fridges with spoilage alerts
- Image classification using CNNs

Finalized idea:

- Transfer learning using a pre-trained CNN model for binary classification (fresh vs rotten)

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

User uploads an image -> Image preprocessed -> CNN model predicts -> Output shown as "Fresh" or "Rotten"

3.2 Solution Requirement

- Dataset: Healthy vs Rotten fruits/vegetables
- Training using Google Colab
- Web deployment using Flask
- Simple UI for image upload and result display

3.3 Data Flow Diagram

Image Upload -> Image Preprocessing -> CNN Model (Transfer Learning) -> Classification Result

3.4 Technology Stack

- Programming Language: Python
- Libraries: TensorFlow/Keras, OpenCV, NumPy, Flask
- Environment: Google Colab
- Frontend: HTML/CSS
- Storage: Google Drive

4. PROJECT DESIGN

4.1 Problem-Solution Fit

Manual inspection methods are outdated and inefficient. A transfer learning-based classification model offers rapid, consistent results and is easy to deploy at scale.

4.2 Proposed Solution

Use a pre-trained model (e.g., MobileNetV2 or VGG16), fine-tuned with a labeled dataset of fresh and rotten fruits and vegetables, to perform binary classification.

4.3 Solution Architecture

Camera/Image Upload -> Preprocessing -> Trained Model -> Predicted Class -> Display on UI

5. PROJECT PLANNING & SCHEDULING

5.1 Timeline

Week 1: Dataset exploration, preprocessing

Week 2: Model development using transfer learning

Week 3: Building Flask app and HTML frontend

Week 4: Testing, bug fixing, and deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Model Accuracy: ~95%
- Testing Dataset: Random samples verified against actual labels
- Real-time Prediction: Flask app delivers accurate and consistent results

7. RESULTS

7.1 Output Screenshots

- UI for uploading an image and receiving prediction
- Displayed output (e.g., "Rotten Tomato", "Fresh Apple")
- Real-time results with image confirmation

8. ADVANTAGES & DISADVANTAGES

Advantages

- Fast and reliable classification
- User-friendly web interface
- Efficient model training using transfer learning

Disadvantages

- Limited to the fruits/vegetables present in the dataset
- Requires GPU for optimal training speed
- Not yet integrated with real-world hardware

9. CONCLUSION

Smart Sorting is a deep learning-powered solution that simplifies the detection of rotten fruits and vegetables. Through the power of transfer learning, the system achieves high accuracy while remaining simple to deploy and use. It addresses key challenges in quality control and waste reduction for the food industry.

10. FUTURE SCOPE

- Expand dataset to include more types of fruits and vegetables
- Real-time deployment using Raspberry Pi and camera modules
- Conveyor belt automation in industrial use cases
- Mobile app version for on-the-go quality checking

11. APPENDIX

Source Code Files

- smart_sorting_model.h5

- class_indices.json

Dataset Link

https://www.kaggle.com/datasets/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten

GitHub Repository & Demo

- GitHub:https://github.com/Anuhya1439/Smart-Sorting-Transfer-Learning-for-Identifying-Rotten-Fruits-and-Vegetables/tree/main/fruitproject

- Demo Video:https://drive.google.com/file/d/1NMGOehId1cob8Fkcni3YDlzeQrwt76pR/view?usp=sharing